Selected Topics in APS Beamline-Controls Development

Tim Mooney

APS Beamline Controls & Data Acquisition

Topics of current interest

- Python
- Cross-platform controls
- Software-distribution
- Burst-integrating scaler
- Fly-scan support

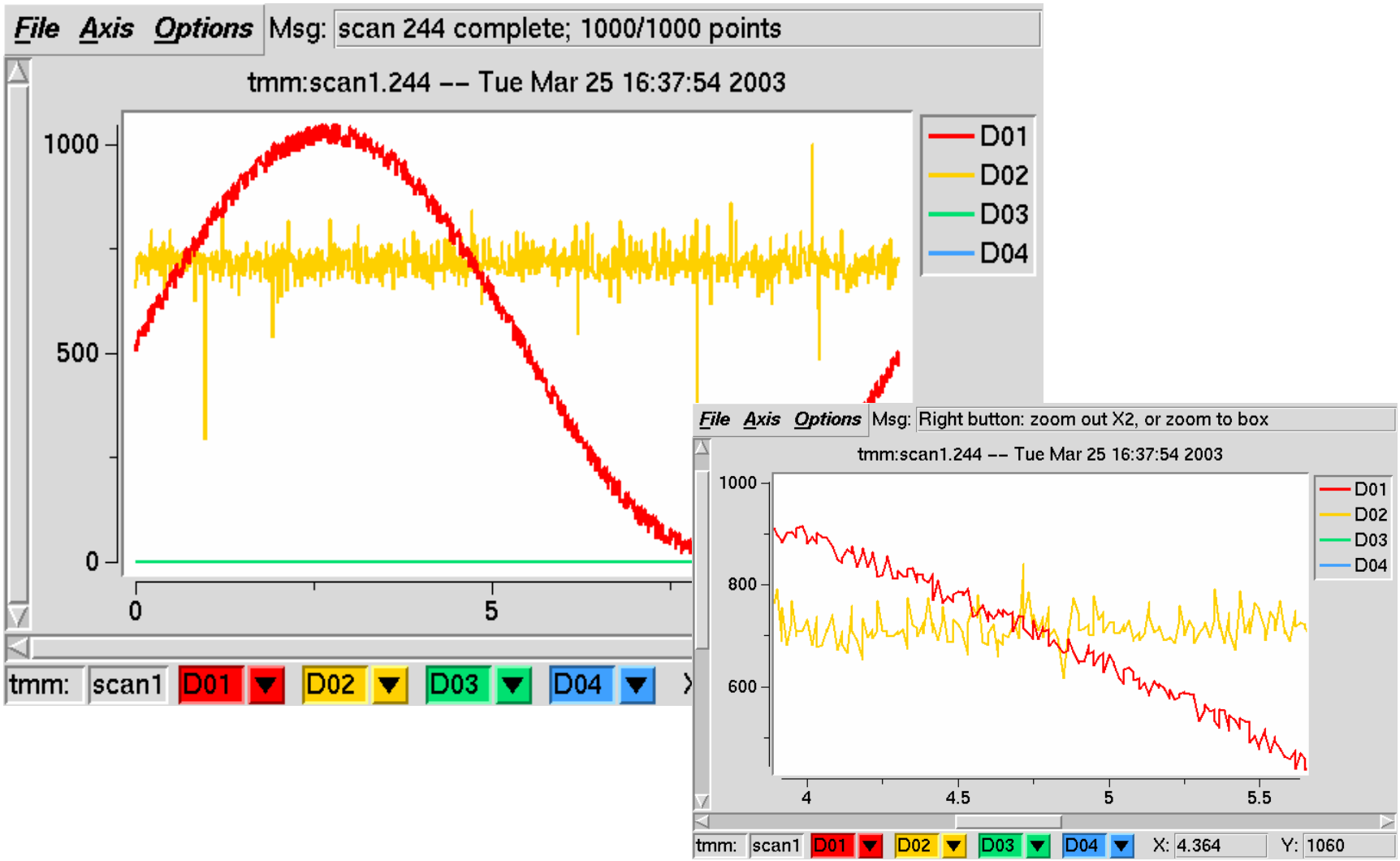
Python: What/Why?

- Embeddable, extensible scripting language
- Object-oriented programming language
- Very good hooks for GUI, math, file I/O
- Good third-party support
- Worth an end user's effort:
  - Suitable for occasional use, little ditties
  - Won't run out of gas as little ditty evolves into serious program

Python: recent work

- EPICS interface to python (Fermilab)
- Beamline status, archive data, control screens, video data via web browser, alarms via email/pager, scan-data plotting, MCA display/analysis (CARS-CAT)
- Run-time data plots, data-file reader, image display (BCDA)
- (www.aps.anl.gov/xfd/bcda/bc_meetings/Welcome.html)

Python examples: plot.py



Python examples: readMDA.py

```
from readMDA import *
dim = readMDA(filename, max_dimension, verbose, help)
```

returns a list:

dim[0] – dictionary of scan-environment variables (mostly PV names)

e.g., **dim[0]['2idd:mca1.R2LO'] = ('ROI_2_LEFT:','',[342])**

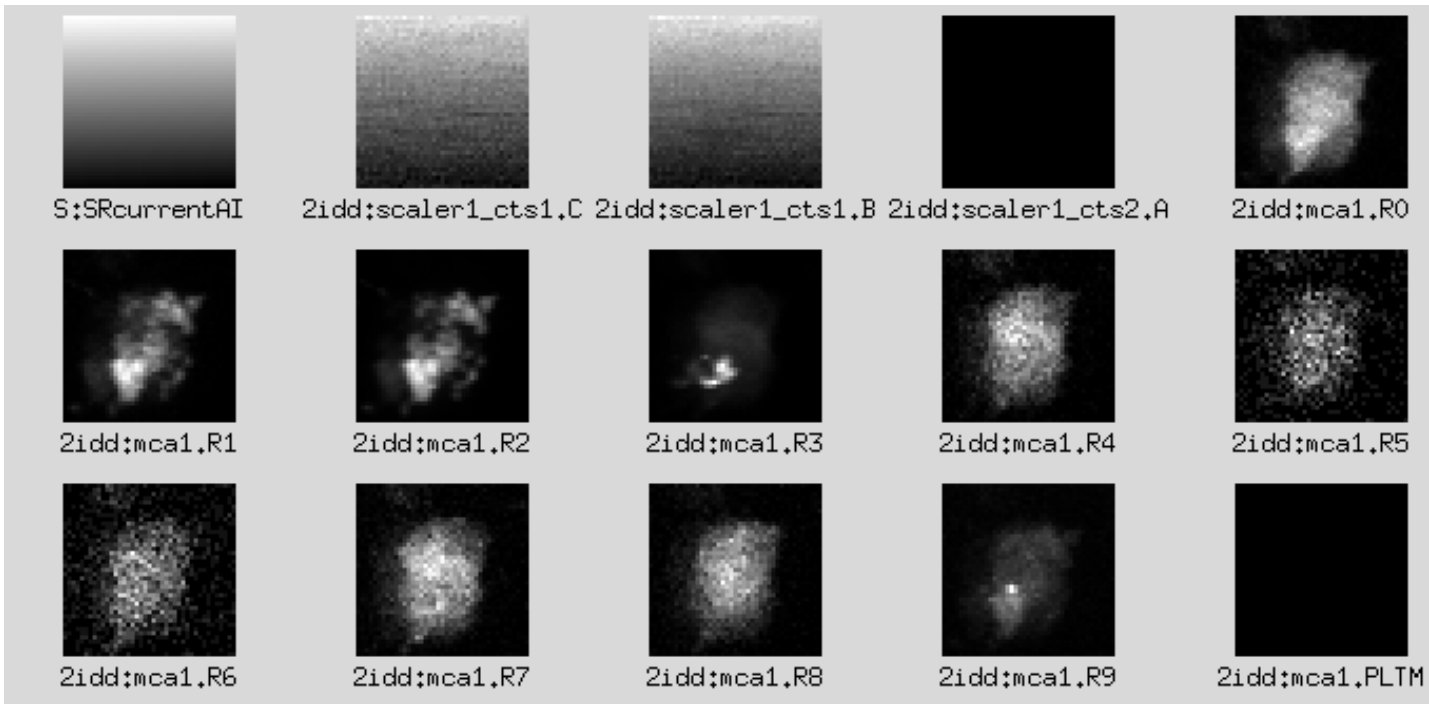
dim[n>0] one dimension of the scan, with the following fields:

time	- date & time at which scan was started
scan_name	- name of scan record that acquired this dimension
curr_pt	- number of data points actually acquired
npts	- number of data points requested
nd	- number of detectors for this scan dimension
d[]	- list of detector-data structures
np	- number of positioners for this scan dimension
p[]	- list of positioner-data structures
nt	- number of detector triggers for this scan dimension
t[]	- list of trigger-info structures

Python examples: detView.py

Requires numeric.py

```
from readMDA import *
from detView import *
d=readMDA("2idd_0004.mda")
detView(d[2].d[0:15],scale=(2,2),columns=5)
```



Cross-platform controls development

- Newest version of EPICS runs (both device side and user side) on vxWorks, Solaris, Linux, Windows, MacOS, RTEMS
- Currently porting beamline-controls code to new version (~1/3 done)
- Targets: niche data-acquisition applications, detector-pool support, CATs who want nothing to do with vxWorks, small labs

New software-distribution model

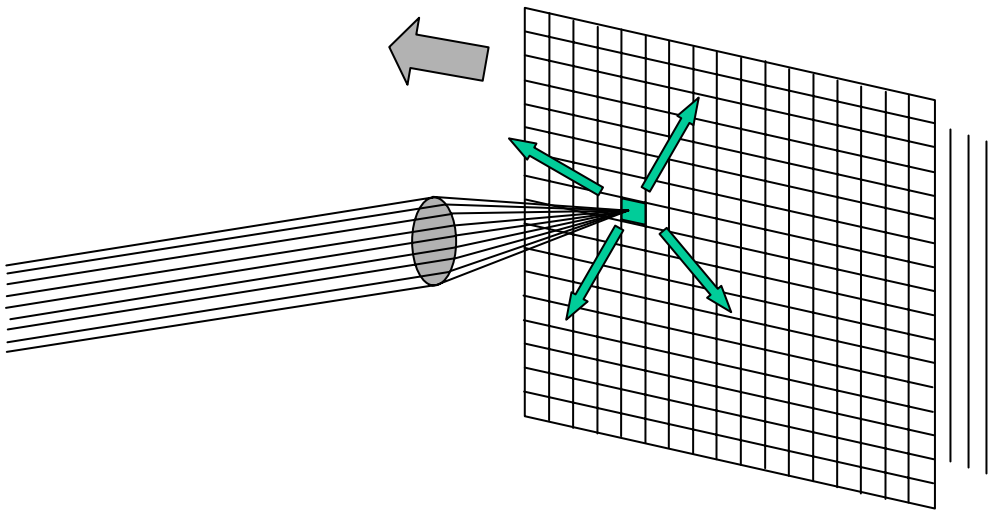
- Previously CATs did this:
  - Install VxWorks, GNU tools
  - Download/build EPICS base
  - Download/build EPICS extensions
  - Download/build synApps (beamline stuff)
  - Build beamline-specific control system
  - Do their real job

New software-distribution model

- Mirrored runnable copies of EPICS, etc. on file server that appears local to beamline
- Now CATs can just do this:
  - Mount one disk
  - Copy example application to local directory
  - Edit one line of one file
  - Build beamline-specific control system
- Can also mirror CAT-developed applications, so CATs have direct access to each other's effort

Fly-scan acquisition strategies

- Microprobe/tomography example



Software-only solution

- Standard software does few-hundred Hz. fly scans, but…
- External coordination of positioner speed with detector-dwell time is required
- Standard motors report position at ~10 Hz.
- Accel./decel. artifacts in raw data
- OK for automated 1D alignment

Waveform Generator + Multichannel Scaler

- Standard solution for scalar data
- Built custom 20-bit digital waveform generator (VME board) for ~100 kHz pixel-rate microscopy/tomography, but…
  - Intense beam can deliver multiple photons per synchrotron burst – can't count edges anymore
  - What about fluorescence?

Multiple photons per SR burst

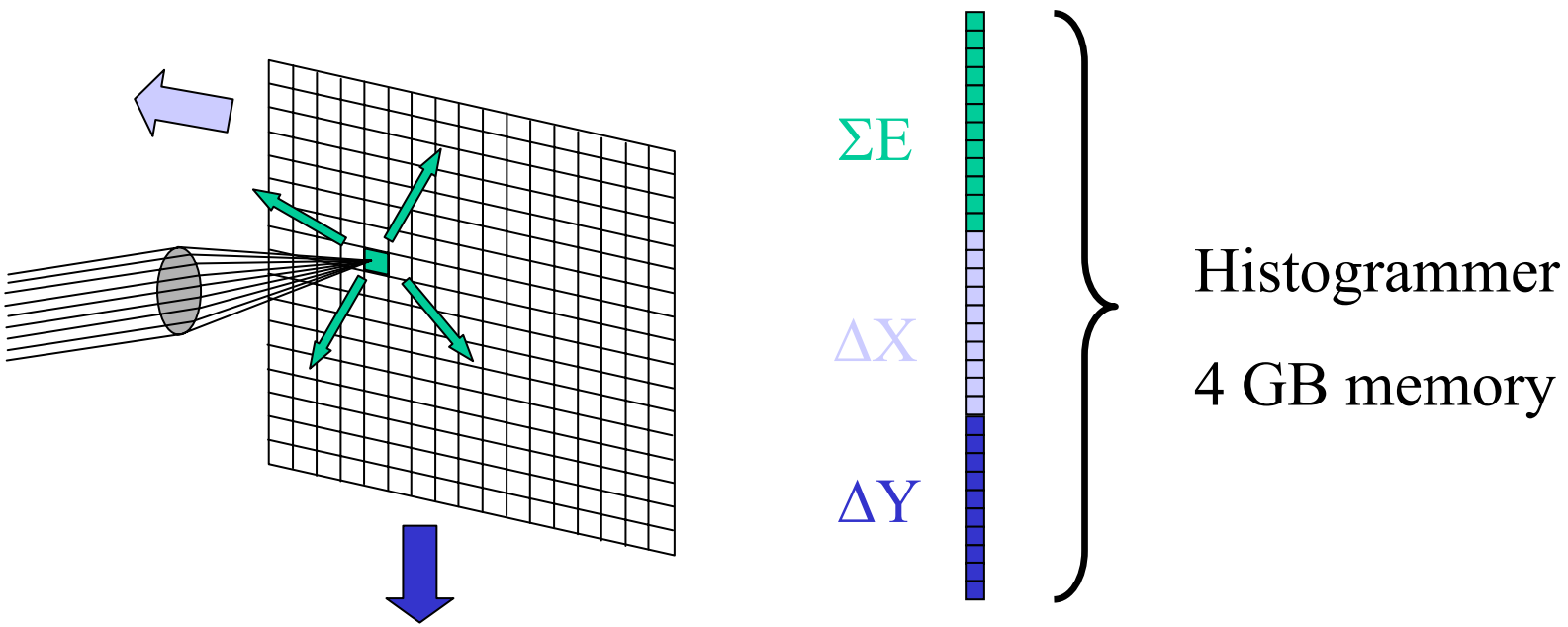
- Integrate/digitize APD output @ burst rate
  - Don't need the excellent differential linearity of a spectroscopy ADC: APD resolution is poor; have sqrt(n) counting statistics anyway.
  - Correlated sampling to remove varying APD background level
  - ~10 ns sample/hold achievable with available analog switches
- Accumulate photon counts digitally
  - Off-the-shelf VME module + some custom engineering

What about fluorescence?

- Can acquire spectra at kHz pixel rates with commercially available histogrammer
- CAMAC or PXI (vendor won't do VME)
- Few running applications

Next-gen microprobe

- Interferometer measures sample ΔX, ΔY, ΔZ
- Few-nm control accuracy?, or…
- Just histogram ΔX, ΔY along with photon energy



Brute-force histogrammer

- Good whenever you can measure something better than you can control it
- Easy to scan image area repeatedly, until adequate statistical precision is achieved
- General-purpose hardware, useful for other kinds of detectors, experiments
- ESRF already has a VME design